

REMARKS

Reconsideration and allowance of the subject application in view of the foregoing amendments and following remarks is respectfully requested.

Claims 1, 4-11, 14-21, 24-31, 34-41, and 44-49 are pending. Claims 1, 4, 10, 11, 14, 17, 20, 21, 24, 27, 30, 31, 34, 37, 40, and 41 have been amended to broaden the scope of the claim language.

Withdrawal of the objection to claim 44 is noted with appreciation.

Withdrawal of the rejection of claims 21-30 under 35 USC 112, second paragraph, as being indefinite is noted with appreciation.

Withdrawal of the rejection of claims 1, 4, 5, 7-11, 14, 15, 17-20, 41, 44, 45, 48, and 49 under 35 USC 102(b) as being anticipated by *Williams* is noted with appreciation.

Claims 1, 4, 5, 7-11, 14, 15, 17-21, 24, 25, 27-35, 37-41, and 44-49 are not obvious over *Williams* (*Mickey Williams, "Microsoft Visual C#.NET"*) in view of GNU "*The GNU C Library, Section 'Explicitly Checking Internal Consistency'*" and further in view of PHP "*(PHP Manual, Section 'assert options')*"

The rejection of claims 1, 4, 5, 7-11, 14, 15, 17-21, 24, 25, 27-35, 37-41, and 44-49 under 35 USC 103(a) as being obvious in view of *Williams* in view of *GNU* and further in view of *PHP* is hereby traversed. Claim 1 is patentable over *Williams* in view of *GNU* and *PHP* because the references, singly or in combination, fail to disclose or suggest every feature of claim 1.

Claim 1

The PTO agrees that *Williams* fails to disclose or suggest "receiving an assertion from an executing process, wherein the executing process is integral to an operating system" as claimed claim 1. The PTO attempts to rely on *GNU* to overcome the admitted deficiency of *Williams*. This is incorrect.

First, the PTO asserts that *GNU* describes "using an assert method in [an] operating system and report execution error (see for example, p. 1, section 'Explicitly Checking Internal Consistency', first paragraph and p.2 third paragraph, 'check for an error return from an operating system function')." See Official Action (OA) mailed January 22, 2008 at page 3, lines 12-17. This is incorrect. The first PTO-identified portion of *GNU*, provided for ease of reference, states:

When you're writing a program, it's often a good idea to put in checks at strategic places for "impossible" errors or violations of basic assumptions. These kinds of checks are helpful in debugging problems with the interfaces between different parts of the program, for example.

GNU at page 1, paragraph 1 of Section "Explicitly Checking Internal Consistency."

The above portion of *GNU* fails to disclose or suggest using an assert method in an operating system and reporting an execution error as claimed in claim 1. The cited portion of *GNU* appears to disclose using checks in programs for debugging problems without disclosing receiving an assertion from an executing process integral to an operating system, recording the assertion, and allowing the executing process (which is integral to the operating system) to continue execution. For at least this reason, withdrawal of the rejection is respectfully requested.

The second PTO-identified portion of *GNU*, provided for ease of reference, states:

Sometimes the "impossible" condition you want to check for is an error return from an operating system function. Then it is useful to display not only where the program crashes, but also what error was returned. The `assert_perror` macro makes this easy.

GNU at page 2, paragraph 3 of Section "Explicitly Checking Internal Consistency."

The above portion of *GNU* fails to disclose or suggest receiving an assertion from an executing process integral to an operating system and reporting an execution error as claimed in claim 1. The cited portion of *GNU* appears to disclose checking for an 'impossible' condition in a program which receives an error return from an operating system function. That is, the cited portion of *GNU* appears to disclose use of the

"assert_perror" macro in a program in order to check for an error returned from an operating system and not use of the macro in the operating system. For at least this reason, withdrawal of the rejection is respectfully requested.

Second, *GNU* appears to disclose that a program aborts without proceeding to continue execution as a result of an assert occurrence. *GNU* at page 2, paragraph 11 of Section "Explicitly Checking Internal Consistency" - "your program should not abort when given invalid input, as assert would do." That is, *GNU* discloses aborting execution upon occurrence and not the claimed receiving an assertion, recording the assertion, and allowing the executing process to continue execution. Additionally, see *GNU* at page 1, paragraph 3, which states that the assert macro "provides a convenient way to **abort the program** while printing a message about where in the program the error was detected." (emphasis added). Thus, the execution is aborted and execution does not continue and a combination of *Williams* and *GNU* fails to render obvious the claimed subject matter. For at least this reason, withdrawal of the rejection is respectfully requested.

Third, the PTO asserts that a person of ordinary skill in the art at the time of the present invention would have been motivated to combine *GNU* with *Williams* in order to "display all the error information and further help to debug the problem as suggested by GNU." OA at page 3, final three lines of the page. This is incorrect and fails to overcome Applicant's description that release code "is typically devoid of assertions because the assertions cause performance degradation." See Instant Specification at paragraph 8. Contrary to the PTO's assertion, *GNU* buttresses this fact at page 1, paragraph 5, reciting that the consistency checks may "make the program significantly slower." Further, Applicants' description recites that "[a]borting . . . continuously running software [i.e., an operating system] results in a system crash, which is an unacceptable artifact of a violated assertion." See Instant Specification at paragraph 8. Thus, the aborting of a program as set forth in *GNU* is an unacceptable occurrence with respect to an operating system. For at least this reason, the PTO has failed to articulate a reasonable rationale for combining the *Williams* and *GNU* as asserted and a prima facie

case of obviousness has not been set forth. For at least this reason, withdrawal of the rejection is respectfully requested.

Fourth, the PTO admits that neither *Williams* nor *GNU* discloses recognizing an assertion request type corresponding to the assertion request. The PTO asserts that *PHP* cures the admitted deficiency of *Williams* and *GNU*. This is also incorrect.

The cited portion of *PHP* appears to describe options useable to control an assertion and fails to disclose an assertion request type as in the present claimed subject matter. The cited portion of *PHP* describes the setting/getting of "various assert flags" related to control options of an assert. Thus, *PHP* does not appear to identify a need nor a disclosure of recognizing an assertion request type as claimed.

Further, as set forth in paragraph 13 of the Instant Specification, being able to differentiate between assertion request types may be beneficial:

[b]ecause different assertion types generally require different amounts of processor resources, the ability to enable only specific types of assertions allows a programmer to better manage the trade-off between the usefulness of a particular type of assertion and its associated cost (in required processor resources)."

For at least this reason, withdrawal of the rejection is respectfully requested.

Fifth, the PTO asserts that *PHP* discloses "determining a component that sourced the assertion request" in p. 1, Table 1. Assert Options. This is incorrect because *PHP* fails to disclose a determination of the component that sourced the assertion request, with or without determining whether the component has assertion requests enabled. *PHP* appears to disclose, as set forth above, setting and getting various flags related to the assert without relation to a particular component and without disclosing determining the component which sourced the assertion request. For at least this reason, withdrawal of the rejection is respectfully requested.

Sixth, the PTO asserts that *Williams* discloses "allowing the executing process to continue execution." This is believed incorrect because although Figure 9-3 depicts a dialog box having an Ignore button (among abort and retry buttons) there is no

supporting description related to operation of the assert method as a result of a user actuating the ignore button (or the abort or retry buttons). The description at page 10 of *Williams* further states only that dialog box displayed as a result of encountering an assertion failure would be "similar to the one shown in Figure 9-3." Thus, *Williams* fails to definitively disclose the contents of the displayed dialog box and, potentially more importantly, fails to disclose what happens as a result of activating one of the displayed buttons. For at least this reason, withdrawal of the rejection is respectfully requested.

Further, combining *GNU* with *Williams* *arguendo* appears to suggest that activating any of the displayed buttons would result in "abort[ing] the program while printing a message about where in the program the error was detected." *GNU* at p. 1, paragraph 3. See also *GNU* at p. 1, paragraph 7 "If it is false (zero), assert aborts the program . . . after printing a message." Thus, the asserted combination of *GNU* and *Williams* appears to suggest aborting a program after assertion receipt and not "allowing the executing process to continue execution" as claimed in claim 1. For at least this reason, withdrawal of the rejection is respectfully requested.

Based on each of the foregoing reasons, amended claim 1 is patentable over *Williams*, alone or in combination with *GNU* and *PHP*, and the rejection is respectfully requested to be withdrawn.

Claims 4, 5, 7-11, 14, 15, 17-21, 24, 25, 27-35, 37-41, and 44-49

Claims 4, 5, 7-11, 14, 15, 17-21, 24, 25, 27-35, 37-41, and 44-49 depend, either directly or indirectly, from claims 1, 11, 21, 31, and 41, include further features, and are patentable over *Williams* in view of *GNU* and further in view of *PHP* for at least the reasons advanced above with respect to claim 1. The rejection of claims 4, 5, 7-11, 14, 15, 17-21, 24, 25, 27-35, 37-41, and 44-49 should be withdrawn.

Claims 6, 16, 26, and 36 are not obvious over Williams in view of GNU, PHP, and Cantrill (US 7,146,473)

The rejection of claims 6, 16, 26, and 36 under 35 USC 103(a) as being obvious over *Williams* in view of *GNU*, *PHP*, and *Cantrill* is hereby traversed. Claims 6, 16, 26, and 36 depend, either directly or indirectly, from claims 1, 11, 21, and 31, include further features, and are patentable over *Williams* in view of *GNU*, *PHP*, and *Cantrill* for at least the reasons advanced above with respect to claims 1, 11, 21, and 31, respectively. The rejection of claims 6, 16, 26, and 36 is respectfully requested to be withdrawn.

Conclusion

All objections and rejections having been addressed, it is respectfully submitted that the present application should be in condition for allowance and a Notice to that effect is earnestly solicited.

To the extent necessary, a petition for an extension of time under 37 C.F.R. 1.136 is hereby made. Please charge any shortage in fees due in connection with the filing of this paper, including extension of time fees, to Deposit Account 08-2025 and please credit any excess fees to such deposit account.

Respectfully submitted,

Jose German RIVERA et al.



Randy A. Noranbrock
Registration No. 42,940
Telephone: (703) 684-1111

HEWLETT-PACKARD COMPANY

IP Administration
Legal Department, M/S 35
P.O. Box 272400
Fort Collins, CO 80528-9599
Telephone: (970) 898-7057
Facsimile: 281-926-7212
Date: **April 10, 2008**
RAN/bjs